INTERNATIONAL STANDARD

ISO/IEC 15414

Third edition
2015-04-15

Information technology — Open distributed processing — Reference model — Enterprise language

Technologies de l'information — Traitement réparti ouvert — Modèle de référence — Langage d'entreprise

Reference number
ISO/IEC 15414:2015(E)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

CONTENTS

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 15414 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*, in collaboration with ITU-T. The identical text is published as ITU-T X.911 (09/2014).

This third edition cancels and replaces the second edition (ISO/IEC 15414:2006), which has been technically revised.

Introduction

The rapid growth of distributed processing led to the adoption of the Reference Model of Open Distributed Processing (RM-ODP). This Reference Model provides a coordinating framework for the standardization of open distributed processing (ODP). It creates an architecture within which support of distribution, interworking and portability can be integrated. This architecture provides a framework for the specification of ODP systems.

The Reference Model of Open Distributed Processing is based on precise concepts derived from current distributed processing developments and, as far as possible, on the use of formal description techniques for specification of the architecture.

This Recommendation | International Standard refines and extends the definition of how ODP systems are specified from the enterprise viewpoint, and is intended for the development or use of enterprise specifications of ODP systems.

0.1 RM-ODP

The RM-ODP consists of:

– Part 1: Rec. ITU-T X.901 | ISO/IEC 10746-1: **Overview**: This contains a motivational overview of ODP, giving scoping, justification and explanation of key concepts, and an outline of the ODP architecture. It contains explanatory material on how the RM-ODP is to be interpreted and applied by its users, who may include standards writers and architects of ODP systems. It also contains a categorization of required areas of standardization expressed in terms of the reference points for conformance identified in ITU-T Rec. X.903 | ISO/IEC 10746-3. This part is informative.

– Part 2: Rec. ITU-T X.902 | ISO/IEC 10746-2: **Foundations**: This contains the definition of the concepts and analytical framework for normalized description of (arbitrary) distributed processing systems. It introduces the principles of conformance to ODP standards and the way in which they are applied. This is only to a level of detail sufficient to support Rec. ITU-T X.903 | ISO/IEC 10746-3 and to establish requirements for new specification techniques. This part is normative.

– Part 3: Rec. ITU-T X.903 | ISO/IEC 10746-3: **Architecture**: This contains the specification of the required characteristics that qualify distributed processing as open. These are the constraints to which ODP standards shall conform. It uses the descriptive techniques from Rec. ITU-T X.902 | ISO/IEC 10746-2. This part is normative.

– Part 4: Rec. ITU-T X.904 | ISO/IEC 10746-4: **Architectural semantics**: This contains a formalization of the ODP modelling concepts defined in Rec. ITU-T X.902 | ISO/IEC 10746-2 clauses 8 and 9. The formalization is achieved by interpreting each concept in terms of the constructs of one or more of the different standardized formal description techniques. This part is normative.

– Rec. ITU-T X.911 | ISO/IEC 15414: **Enterprise language**: this Recommendation | International Standard.

0.2 Overview and motivation

Part 3 of the Reference Model, Rec. ITU-T X.903 | ISO/IEC 10746-3, defines a framework for the specification of ODP systems comprising:

1) five viewpoints, called enterprise, information, computational, engineering and technology, which provide a basis for the specification of ODP systems;

2) a viewpoint language for each viewpoint, defining concepts and rules for specifying ODP systems from the corresponding viewpoint.

The purpose of this Recommendation | International Standard is to:

– Refine and extend the enterprise language defined in Rec. ITU-T X.903 |ISO/IEC 10746-3 to enable full enterprise viewpoint specification of an ODP system.

– Explain the correspondences of an enterprise viewpoint specification of an ODP system to other viewpoint specifications of that system.

– Ensure that the enterprise language, when used together with the other viewpoint languages, is suitable for the specification of a concrete application architecture to fill a specific business need.

This Recommendation | International Standard uses concepts taken from Recs ITU-T X.902 | ISO/IEC 10746-2 and X.903 | ISO/IEC 10746-3 and structuring rules taken from clause 5 of Rec. ITU-T X.903 | ISO/IEC 10746-3; it introduces refinements of those concepts, additional viewpoint-specific concepts, and prescriptive structuring rules for enterprise viewpoint specifications. The additional viewpoint-specific concepts are defined using concepts from Recs ITU-T X.902 | ISO/IEC 10746-2 and X.903 | ISO/IEC 10746-3.

This Recommendation | International Standard provides a common language (set of terms and structuring rules) to be used in the preparation of an enterprise specification capturing the purpose, scope and policies for an ODP system. An enterprise specification is a part of the specification of an ODP system using viewpoints defined by Recommendation ITU-T X.903 | ISO/IEC 10746-3. The specification of the ODP system can describe any or all of:

- an existing system within its environment;
- an anticipated future structure or behaviour of that existing system within an existing or an anticipated future environment;
- a system to be created within some environment.

The primary audience for this Recommendation | International Standard is those who prepare and use such specifications. The audience includes ODP system owners and users, including subject management experts, and developers and maintainers of ODP systems, tools and methodologies.

The motivation for the enterprise language is to support standardized techniques for specification. This improves communication and helps create consistent specifications.

The preparation of specifications often falls into the category referred to as analysis or requirement specification. There are many approaches used for understanding, agreeing and specifying systems in the context of the organizations of which they form a part. The approaches can provide useful insights into both the organization under consideration and the requirements for systems to support it, but they generally lack the rigour, consistency and completeness needed for thorough specification. The audiences of the specifications also vary. For agreement between the potential users of an ODP system and the provider of that system, it may be necessary to have different presentations of the same system – one in terms understood by clients, and one in terms directly related to system realization.

The use of enterprise specifications can be wider than the early phases of the software engineering process. A current trend is to integrate existing systems into global networks, where the functionality of interest spans multiple organizations. The enterprise language provides a means to specify the joint agreement of common behaviour of the ODP systems within and between these organizations. The enterprise specification can also be used in other phases of the system life cycle. The specification can, for example, be used at system run-time to control agreements between the system and its users, and to establish new agreements according to the same contract structure. Enterprise viewpoint specifications may contain rules for inter-organizational behaviour.

This Recommendation | International Standard also provides a framework for the development of software engineering methodologies and tools exploiting ODP viewpoint languages, and a set of concepts for the development of enterprise viewpoint specification languages. For these purposes, this Recommendation | International Standard provides rules for the information content of specifications and the grouping of that information. Further requirements on the relationships between enterprise language concepts and concepts in other viewpoints are specific to the methodologies, tools or specification languages to be developed.

An enterprise specification defines the purpose, scope, and policies of an ODP system and it provides a statement of conformance for system implementations. The purpose of the system is defined by the specified behaviour of the system while policies capture further restriction on the behaviour between the system and its environment or within the system itself related to the business decisions by the system owners.

An enterprise specification also allows the specification of an ODP system that spans multiple domains and is not owned by a single party, and specification of the collective behaviour of a system that is divided into independently specified and independently working subsystems.

This generality places greater emphasis on the expression of correct or normal behaviour and on the chains of responsibility involved in achieving it. For example, the advent of service oriented and cloud computing has led to the need to specify business rules and behaviour in a way that clearly describes obligations, permissions, authorizations and prohibitions, as well as the accountability of each of the objects involved in an enterprise specification. This involves the expression of the so-called deontic aspects of the behaviour of the system, and of the accountability of the objects involved.

Annex A presents a metamodel of the enterprise language, illustrating the key concepts of the enterprise language and their relationships. This annex is normative. Annex B provides examples using the concepts and structuring rules of the enterprise language and provides examples of how they may be used. Annex C indicates how the semantics of deontic constraints may be expressed. Annexes B and C are informative.

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

Information technology – Open distributed processing – Reference model – Enterprise language

1 Scope

This Recommendation | International Standard provides:

a) a language (the enterprise language) comprising concepts, structures, and rules for developing, representing and reasoning about a specification of an ODP system from the enterprise viewpoint (as defined in Rec. ITU-T X.903 | ISO/IEC 10746-3);

b) rules which establish correspondences between the enterprise language and the other viewpoint languages (defined in Rec. ITU-T X.903 | ISO/IEC 10746-3) to ensure the overall consistency of a specification.

The language is specified to a level of detail sufficient to enable the determination of the compliance of any modelling language to this Recommendation | International Standard and to establish requirements for new specification techniques.

This Recommendation | International Standard is intended for use in preparing enterprise viewpoint specifications of ODP systems, and in developing notations and tools to support such specifications.

As specified in clause 5 of Rec. ITU-T X.903 | ISO/IEC 10746-3, an enterprise viewpoint specification defines the purpose, scope and policies of an ODP system.

This Recommendation | International Standard is a refinement and extension of Rec. ITU-T X.903 | ISO/IEC 10746-3, clauses 5 and 10, but does not replace them.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical ITU-T Recommendations | International Standards

– Recommendation ITU-T X.902 (2009) | ISO/IEC 10746-2:2010, *Information technology – Open Distributed Processing – Reference Model: Foundations*.

– Recommendation ITU-T X.903 (2009) | ISO/IEC 10746-3:2010, *Information technology – Open Distributed Processing – Reference Model: Architecture*.

– Recommendation ITU-T X.904 (1997) | ISO/IEC 10746-4:1998, *Information technology – Open Distributed Processing – Reference Model: Architectural semantics*.

– Recommendation ITU-T X.906 (1997) | ISO/IEC 19793:2012, *Information technology – Open distributed processing – Use of UML for ODP system specifications.*

2.2 Additional References

– ISO/IEC 19505-2:2012, *Information Technology – Object Management Group Unified Modelling Language (OMG UML) – Part 2: Superstructure*.

3 Terms and definitions

3.1 Definitions from ODP standards

3.1.1 Modelling concept definitions

This Recommendation | International Standard makes use of the following terms as defined in Rec. ITU-T X.902 | ISO/IEC 10746-2.

- action;
- activity;
- behaviour (of an object);
- composite object;
- composition;
- configuration (of objects);
- conformance;
- conformance point;
- contract;
- <X> domain;
- entity;
- environment contract;
- environment (of an object);
- epoch;
- establishing behaviour;
- event;
- instantiation (of an <X> template);
- internal action;
- invariant;
- liaison;
- location in time;
- name;
- object;
- obligation;
- ODP standards;
- ODP system;
- permission;
- policy;
- policy declaration;
- policy envelope;
- policy setting behaviour;
- policy value;
- prohibition;
- proposition;
- reference point;
- refinement;
- role;
- service;
- state (of an object);
- subsystem;

- subtype;
- system;
- <X> template;
- terminating behaviour;
- type (of an <X>);
- viewpoint (on a system).

3.1.2 Viewpoint language definitions

This Recommendation | International Standard makes use of the following terms as defined in Rec. ITU-T X.903 | ISO/IEC 10746-3.

- binder;
- capsule;
- channel;
- cluster;
- community;
- computational behaviour;
- computational binding object;
- computational object;
- computational interface;
- computational viewpoint;
- dynamic schema;
- engineering viewpoint;
- enterprise object;
- enterprise viewpoint;
- <X> federation;
- information object;
- information viewpoint;
- interceptor;
- invariant schema;
- node;
- nucleus;
- operation;
- protocol object;
- static schema;
- stream;
- stub;
- technology viewpoint;
- <viewpoint> language.

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply:

ODP	Open Distributed Processing
RM-ODP	Reference Model of Open Distributed Processing (Recs ITU-T X.901 to X.904 | ISO/IEC 10746 Parts 1-4)

5 Conventions

This Recommendation | International Standard contains references to Parts 2 and 3 of the RM-ODP and to the normative text of this Recommendation | International Standard. Each reference is of one of these forms:

- [Part 2-n.n] – a reference to clause n.n of RM-ODP Part 2: Foundations, X.902 | ISO/IEC 10746-2;
- [Part 3-n.n] – a reference to clause n.n of RM-ODP Part 3: Architecture, X.903 | ISO/IEC 10746-3;
- [n.n] – a reference to clause n.n of this Recommendation | International Standard.

For example, [Part 2-9.4] is a reference to Part 2 of the reference model, (Rec. ITU-T X.902 | ISO/IEC 10746-2), clause 9.4 and [6.5] is a reference to clause 6.5 of this Recommendation | International Standard. These references are for the convenience of the reader.

This Recommendation | International Standard also contains some text which is a modification of text from Part 3 of the reference model, Rec. ITU-T X.903 | ISO/IEC 10746-3. Such text is marked by a reference like this: [see also 3-5.n]. The modifications are authoritative with respect to the enterprise language.

6 Concepts

The concepts of the enterprise language defined in this Recommendation | International Standard comprise:

- the concepts identified in clauses 3.1.1 and 3.1.2 as they are defined in Rec. ITU-T X.902 | ISO/IEC 10746-2 and in ITU-T X.903 | ISO/IEC 10746-3;
- the concepts defined in this clause.

The grouping into subclauses and the headings of the subclauses of this clause are informative.

6.1 System concepts

6.1.1 scope (of a system): The behaviour that a system is expected to exhibit.

6.1.2 field of application (of a specification): The properties the environment of the ODP system shall have for the specification of that system to be used.

6.2 Community concepts

6.2.1 objective (of an <X>): Practical advantage or intended effect, expressed as preferences about future states.

NOTE 1 – Some objectives are ongoing, some are achieved once met.

NOTE 2 – In the text of Rec. ITU-T X.903 | ISO/IEC 10746-3 [Part 3-5] the terms *purpose* and *objective* are synonymous. The enterprise language emphasizes the term *objective* and emphasizes the need to express an objective in measurable terms.

6.2.2 community object: A composite enterprise object that represents a community. The components of a community object are objects of the community represented.

6.3 Behaviour concepts

6.3.1 active enterprise object: An enterprise object that is able to fill an action role. In other words, it is an enterprise object that can be involved in some behaviour.

NOTE – The behaviour of active enterprise objects is constrained by deontic and accountability concepts, defined in clauses 6.4 and 6.6. The deontic tokens defined in clause 6.4 are not themselves active enterprise objects.

6.3.2 actor (with respect to an action): A role (with respect to that action) in which the enterprise object fulfilling the role participates in the action. That object may be called an actor.

NOTE – It may be of interest to specify which actor initiates that action.

6.3.3 artefact (with respect to an action): A role (with respect to that action) in which the enterprise object fulfilling the role is referenced in the action. That object may be called an artefact.

NOTE 1 – An enterprise object that is an artefact in one action can be an actor in another action.

NOTE 2 – The object filling an artefact role in an action is an active enterprise object being referenced in the action and this should not be confused with the way a deontic token held by an object involved in the action constrains its performance.

6.3.4 resource (with respect to an action): A role (with respect to that action) in which the enterprise object fulfilling the role is essential to the action, requires allocation, or may become unavailable. That object may be called a resource.

NOTE 1 – Allocation of a resource object may constrain other behaviours for which that resource is essential.

NOTE 2 – A consumable resource object may become unavailable after some amount of use. Any resource object may become unavailable after some amount of time (for example, if a duration or expiry time has been specified for the resource).

6.3.5 **interface role**: A role in a community, identifying behaviour which takes place with the participation of objects that are not members of that community.

6.3.6 **process**: A collection of steps taking place in a prescribed manner.

NOTE 1 – The prescribed manner may be a partially ordered sequence of steps.

NOTE 2 – The activity structure concepts provided in clause 13.1 of Rec. ITU-T X.902 | ISO/IEC 10746-2 may be used, after substitution of 'step' for 'action' and 'process' for 'activity', to specify the structure of a process.

NOTE 3 – A process may have multiple end points.

NOTE 4 – An enterprise specification may define types of process and may define process templates.

NOTE 5 – A process is an abstraction of a behaviour, and so shares any objectives defined for that behaviour.

NOTE 6 – A process specification can be a workflow specification.

6.3.7 **step**: An abstraction of an action, used in a process, that may leave unspecified some or all of the objects that participate in that action.

6.3.8 **violation**: A behaviour contrary to that required by a rule.

NOTE – A rule or policy may provide behaviour which is to occur upon violation of that, or some other, rule or policy.

6.4 Deontic concepts

6.4.1 **deontic token**: An enterprise object which expresses a constraint on the ability of an active enterprise object holding it to perform certain actions. An active enterprise object carries a set of deontic tokens, which control the occurrence of conditional actions within its behaviour. These tokens are either permits, burdens or embargos. A deontic token is not itself an active enterprise object; it is held by exactly one active enterprise object.

NOTE – The constraint is expressed by a rule forming part of the token; an appropriate notation for expressing this rule will be selected by the specifier. The notation allows the declaration of the active enterprise object and conditional action to which it applies, and requirements on other enterprise objects fulfilling roles in the conditional action controlled. For example, the rule may control the performance of a purchase action by a consumer, and place restrictions on the supplier and the artefact being purchased. The notation may also declare periods of validity or deadlines for performance of the action. The kind of associated information allowed will depend on whether the token is a permit, a burden or an embargo.

6.4.2 **token group**: A group of tokens named so that it can be referred to as a whole.

NOTE – A notation for expressing deontic rules will provide the means for declaring and naming groups of deontic tokens. Changes that result, for example, from the performance of speech acts can then be applied to complete groups of tokens without the need to reference all the group members individually.

6.4.3 **burden**: A deontic token encapsulating the statement of an obligation on the active enterprise object holding it, thereby modifying the urgency of the active enterprise object in performing associated conditional actions within its behaviour.

6.4.4 **embargo**: A deontic token encapsulating the statement of a prohibition on the active enterprise object holding it, thereby modifying the ability of the active enterprise object to perform associated conditional actions within its behaviour.

6.4.5 **permit**: A deontic token encapsulating the statement of a permission on the active enterprise object holding it, thereby modifying the ability of the active enterprise object to perform associated conditional actions within its behaviour.

6.4.6 **conditional action**: An action which has associated preconditions based on the sets of burdens, permits and embargos carried by the active enterprise objects filling its various action roles. The specification of the conditional action states what permits are required for, what burdens favour, and what embargos inhibit performance of the action.

6.4.7 **speech act**: An action whose performance results in a change to the sets of deontic tokens (permits, embargos and burdens) carried by the active enterprise objects filling its various action roles. A speech act may result in the addition of new tokens to the performer of an action role, or in the removal of tokens from the performer of an action role, or the transfer of tokens from the performer of one action role to the performer of another action role in the same interaction.

NOTE 1 – Many actions for which parties are accountable are speech acts; examples are prescription and commitment.

NOTE 2 – Although we speak informally of a speech act as changing or transferring a token, it is more precise to describe this process as the destruction of one of the token existing before the act occurred and the construction of a new token based on the information available when the act is performed. The definition of the speech act type includes the formal rules governing the content and location of the token that is generated. Transfer of a token by a speech act is therefore the process of destruction of a token held by one of the participating objects followed by construction of a new token with the same contents at its destination.

6.5 Policy concepts

6.5.1 **policy**: A constraint on a system specification foreseen at design time, but whose detail is determined subsequent to the original design, and capable of being modified from time to time in order to manage the system in changing circumstances. A policy is expressed as a rule, which may, in turn, be a composition of several sub-rules. A policy is introduced into a specification by a policy declaration. At any point in time it has a particular policy value, but the policy value can be changed by a defined policy setting behaviour, so long as it remains within a defined policy envelope. [See 2-11.2.8 to 2-11.2.12]

NOTE – A rule can be expressed as an obligation, an authorization, a permission or a prohibition. Not all the constraints involved restrict the behaviour; for example, some policies may represent an empowerment.

6.5.2 **affected behaviour**: A fragment of behaviour (including an action, step or process) that is constrained by the current policy value.

6.6 Accountability concepts

6.6.1 **party**: An enterprise object modelling a natural person or any other entity considered to have some of the rights, powers and duties of a natural person.

NOTE 1 – Examples of parties include enterprise objects representing natural persons, legal entities, governments and their parts, and other associations or groups of natural persons.

NOTE 2 – Parties are responsible for their actions and the actions of their agents.

The following concepts are used to identify actions which involve the accountability of a party.

6.6.2 **commitment**: An action resulting in an obligation by one or more of the participants in the act to comply with a rule or perform a contract.

NOTE 1 – The enterprise objects participating in an action of commitment may be parties or agents acting on behalf of a party or parties. In the case of an action of commitment by an agent, the principal responsible for the agent becomes obligated.

NOTE 2 – The fact that an enterprise object is obligated is expressed by associating with it a burden describing the obligation.

6.6.3 **prescription**: An action that establishes a rule.

6.6.4 **authorization**: An action indicating that a particular behaviour shall not be prevented.

NOTE 1 – Unlike a permission, an authorization is an empowerment.

NOTE 2 – The fact that an enterprise object has performed an authorization is expressed by it issuing a required permit and itself undertaking a burden describing its obligation to facilitate the behaviour.

6.6.5 **declaration**: An action that establishes a state of affairs in the environment of the object making the declaration.

NOTE – The essence of a declaration is that, by virtue of the act of declaration itself and the authorization of the object making the declaration or its principal, the declaration action causes a state of affairs to come into existence outside that object.

6.6.6 **delegation**: The action that assigns something, such as authorization, responsibility or provision of a service to another object.

NOTE – A delegation, once made, may later be withdrawn.

6.6.7 **evaluation**: An action that assesses the value of something.

NOTE 1 – For example, the action by which an ODP system assigns a relative status to a thing, according to estimation by the system.

NOTE 2 – Value can be considered in terms of usefulness, importance, preference, acceptability, etc.; the evaluated target may be, for example, a credit rating, a system state, a potential behaviour, etc.

6.6.8 **agent**: An active enterprise object that has been delegated something (authorization, responsibility, provision of a service, etc.) by, and acts for, a party (in exercising the authorization, carrying out the responsibility, providing the service, etc.).

NOTE 1 – An agent may be a party or may be the ODP system or one of its components. Another system in the environment of the ODP system may also be an agent of a party.

NOTE 2 – The delegation may have been direct, by a party, or indirect, by an agent of the party having authorization from the party to so delegate.

NOTE 3 – A specification may state that, in its initial state, an active enterprise object is an agent of a party.

6.6.9 **principal**: A party that has delegated something (authorization, provision of a service, etc.) to another.

7 Structuring rules

This clause refines and extends the structuring rules defined in clause 5.2 of Rec. ITU-T X.903 | ISO/IEC 10746-3, as they apply to the concepts of community, enterprise object, objective, behaviour and policy. It defines structuring rules for the accountability concepts defined in clause 6.6. It uses the concepts defined in ITU-T Rec. X.902 | ISO/IEC 10746-2, in clause 5.1 of Rec. ITU-T X.903 | ISO/IEC 10746-3 and in clause 6.

7.1 Overall structure of an enterprise specification

An enterprise specification of an ODP system is a description of that system and relevant parts of its environment. The enterprise specification focuses on the scope and purpose of that system and the policies that apply to it in the context of its environment.

NOTE 1 – The environment of an ODP system and the ODP system itself may span multiple organizations. More than one party may own the ODP system.

NOTE 2 – An enterprise specification may specify the collective behaviour of separately specified and interworking subsystems of the ODP system.

A fundamental structuring concept for enterprise specifications is that of a community. A community is a configuration of enterprise objects that describes a collection of entities (e.g., human beings, information processing systems, resources of various kinds and collections of these) that is formed to meet an objective. These entities are subject to an agreement governing their collective behaviour. The assignment of actions to the enterprise objects that comprise a community is defined in terms of roles. (See clauses 7.8.1 and 7.8.2.)

The enterprise specification includes, within the areas of interest of the specification users, the objective and scope of the ODP system, the policies for the ODP system (including those of any environment contracts), the community in which the ODP system is specified and the roles fulfilled by the ODP system and other enterprise objects in that community, and the processes in which the ODP system and enterprise objects in its environment participate.

An enterprise specification of an ODP system includes at least the community in which that system may be represented as a single enterprise object interacting with its environment. Whether the specification actually includes more than that level of abstraction is left for the specifier to decide.

NOTE 3 – This minimal enterprise specification describes the objective and scope of the ODP system; this description is necessary for completeness of the enterprise specification.

Where necessary for clarity or completeness of description of the behaviour of the ODP system, the enterprise specification can include any other communities of which the ODP system or its components are members, and other communities of which enterprise objects in the environment of the ODP system are members.

NOTE 4 – The set of communities in an enterprise specification may include, for example, communities at both more abstract and more detailed levels than the minimal enterprise specification, as well as communities relating to functional decomposition of the ODP system and to ownership of the ODP system and its parts.

The enterprise specification can also be structured in terms of a number of communities interacting with each other.

NOTE 5 – This may represent, for example, a federation.

The scope of the system is defined in terms of its intended behaviour; in the enterprise language this is expressed in terms of roles, processes or policies, and the relationships of them. Reference to behaviour in general includes both basic behaviour, in terms of processes, steps and actions, and any associated deontic or accountability mechanisms.

NOTE 6 – It may be meaningful to discuss the intended, delivered or expected scope of a system in various phases of planning, development or deployment. In such cases, the term "scope" should be appropriately qualified.

A complete ODP system specification indicates rules for internal consistency in terms of relationships between various viewpoint specifications and a complete enterprise specification contains conformance rules that define the required behaviour of the described ODP system.

7.2 Contents of an enterprise specification

An enterprise specification is structured in terms of the elements explained in clause 7.1 and the other concepts identified in clause 6, as well as the relationships between them.

For each of these elements, depending on the specifier's choice and desired level of detail, the enterprise specification provides:

- the characteristics of the element; or
- the type or types of the element; or
- a template for the element.

An enterprise specification provides a pattern for realization of an ODP system in its environment. As such it may be realized once, never, or many times, depending upon the objective of the specifier. This means that the behaviour defined may also be observable any number of times, depending on when and where the specification is realized. It is therefore necessary to take care of the context when interpreting statements about the occurrence of the concepts in an enterprise specification.

In particular, when distinguishing type and occurrence in a specification, the aim is normally to distinguish between multiple occurrences of a single type within the specification, and not to imply a constraint on how often the specification can be realized in the world. The definitions in this Recommendation | International standard should be interpreted in the context of a specification, without constraining when and where the specification should be realized.

The enterprise language makes no prescription about the specification process nor about the level of abstraction to be used in an enterprise specification.

NOTE 1 – No recommendations are made about the relative merits of modelling from top-down or bottom-up. Nor is there a recommended sequencing of the development of viewpoint specifications.

NOTE 2 – It is a design choice whether a specification deals with a specific implementation by, for example, identifying individual enterprise objects, or deals with a more flexible architecture by identifying types and rules for assigning enterprise objects to roles.

NOTE 3 – A specification may be partitioned because of readability, reuse of specification fragments in other specifications or interoperability of enterprise objects.

NOTE 4 – Roles and communities, as well as types and templates, can be private to a specification and development environment, or they can be stored in a repository that can be shared by a wider audience involving several development environments and groups.

7.3 Community rules

7.3.1 Community

An enterprise specification states the objective of a community, how it is structured, what it does, and what objects comprise it. The objective of the community is expressed in a contract that specifies how the objective can be met. This contract:

- states the objective for which the community exists;
- governs the structure, the behaviour and the policies of the community;
- constrains the behaviour of the members of the community;
- states the rules for the assignment of enterprise objects to roles.

A community contract constrains the existence or behaviour of its enterprise object members in such a way as to satisfy the objectives of the community. The community objective may be decomposed into sub-objectives and the behaviour decomposed into distinct sub-behaviours that each satisfies one of the sub-objectives.

In general, the enterprise specification defines the behaviour which sets up the community contracts and each contract specifies the behaviour necessary to terminate it. However, a specification may cover only a certain period of time throughout which the community and its contract exist. In such cases, their existence forms part of the initial state of the specification, creation being implicit.

An enterprise specification may reuse an existing contract template, incorporating it as a specification fragment by reference. In some cases, the contract may define new enterprise objects that are to be created as part of the community creation, or the community may be formed from pre-existing enterprise objects. The creation of a new community will, in general, place obligations, permissions and prohibitions on its founder members.

The collective behaviour of the community is specified in terms of one or more of the following elements:

- the roles of the community (including those roles which define how a community interacts with its environment);
- the processes that take place in the community;
- the assignment of roles to steps in processes;
- policies that apply to the roles and processes;
- the allocation and manipulation of deontic tokens that constrain the actions, steps or processes; and
- identification of those actions for which parties assigned roles are accountable.

This collective behaviour is constrained by the policies associated with roles and processes and by the contract of the community.

The behaviours of objects in a community are subject to the contract of that community and to the constraints specified in relationships between those objects.

The community is further defined in terms of the following elements:

- roles;
- rules and policies for assignment of enterprise objects to roles;
- relationships between roles;
- relationships of roles to processes;
- rules and policies that apply to roles and to relationships between roles;
- rules and policies that apply to relationships between enterprise objects in the community;
- behaviour that changes the structure or the members of the community during the lifetime of that community.

NOTE 1 – Types of communities or community templates may be used in the specification of a community.

NOTE 2 – Types of communities may be related by refinement.

NOTE 3 – A family of related contracts may be generated from a contract template. Some aspects of the contract (e.g., membership) may only apply to particular instantiations of the contract template, while other aspects may apply to all instantiations of the contract template. For example, assignment rules and policies can be considered as parameters in a contract template. The style of contract specification determines the method of community establishment, as well as other aspects of the community life cycle.

NOTE 4 – The specification of a community may include specific enterprise objects, relationships between those objects, and relationships of those object to enterprise objects assigned to roles in that community.

NOTE 5 – The concept of a contract is particularly important within a community specification, because the contract contains all the information about the structure of a community, its behaviour, and the way it operates.

7.3.2 Relationships between communities

An enterprise specification can include one or more communities. Interactions between active enterprise objects fulfilling appropriate roles within different communities can be considered as interactions between those communities.

When composing *communities*, there will be a set of *policies* common to those *communities*. These *policies* shall be consistent, although unspecified *behaviour* in the composite *community* may allow room for (mutually inconsistent) *behaviour* in each individual *community*.

Communities may interact in the following ways:

- a community object fulfils one or more roles in other communities;
- two or more community objects interact in fulfilling roles in some other community;
- the enterprise specification requires the same object to fulfil specific roles in more than one community and the behaviour of the object in any given role may affect its behaviour in other roles;
- an object, in fulfilling an interface role (see clause 7.8.3) of one community, interacts with an object fulfilling an interface role in another community;
- a community includes behaviour for creating new communities.

NOTE 1 – For example, federation establishment means creation of a new community, which involves putting in place the contract of the community, including the structure and policies for that community.

NOTE 2 – For more information about interactions involving community objects and the communities they represent, see clause 7.8.3 – Interface roles and interactions between communities.

For each of these ways of interacting there is an invariant that determines the constraints on the collective behaviour of the communities concerned. In all kinds of interactions of communities it is critical to consider the invariants that determine the constraints on the collective behaviour of the communities concerned, and the objectives and policies that govern the different communities. The communities involved in an interaction may have differing rules; all of the objects participating in that interaction must be able to conform to all those rules.

These invariants include:

- where a community object fulfils one or more roles in another community, the community that the community object represents is governed by the policies of the other community;
- where two or more community objects interact in fulfilling roles in some other community, the communities that the community objects represent are related by those interactions;
- where the same object is required to fill specific roles in more than one community, an invariant specifies how the actions of that object affect those communities;